DISTRIBUTED AND INTERACTIVE SIMULATIONS OPERATING AT LARGE SCALE FOR TRANSCONTINENTAL EXPERIMENTATION

Thomas D. Gottschalk
Center for Advanced Computing Research
California Institute of Technology
Pasadena California USA
tdg@cacr.caltech.edu

Ke-Thia Yao, Gene Wagenbreth,
Robert F. Lucas & Dan M. Davis
Information Sciences Institute
University of Southern California
Marina del Rey, California USA
{kyao, genew, rflucas, ddavis} @isi.edu

Abstract— This paper addresses the use of emerging technologies to respond to the increasing needs for larger and more sophisticated agent-based simulations of urban areas. The U.S. Joint Forces Command has found it useful to seek out and apply technologies largely developed for academic research in the physical sciences. The use of these techniques in transcontinentally distributed, interactive experimentation has been shown to be effective and stable and the analyses of the data find parallels in the behavioral sciences. The authors relate their decade and a half experience in implementing high performance computing hardware, software and user inter-face architectures. These have enabled heretofore unachievable results. They focus on three advances: the use of general purpose graphics processing units as computing accelerators, the efficiencies derived from implementing interest managed routers in distributed systems, and the benefits of effective data management for the voluminous information.

Keywords-component; High Performance Computing, GPGPUs, software routers;10Gig Networking

I. INTRODUCTION

This paper addresses the authors' experiences with three new technologies: 1) A new GPU accelerator-enhanced Linux Cluster 2) A trans-continental test of 10 Gigabit WANs using interest-managed routers 3) An optimized distributed data management scheme

II. DOD GOALS AND OBJECTIVES

JFCOM's mission is to lead the transformation of the Armed Forces into the 21st Century via their Joint Concept Development and Experimentation Directorate, J9. This mandate calls for experiments with war-fighters staffing the consoles during interactive simulations. The J9 codes consist of representations of terrain that are populated with intelligent-agent friendly forces, enemy forces and civilian groups. JFCOM required simulations of more than 2,000,000 entities on a global-scale terrain database [1]. The line-of-sight calculations between the entities are an "n-squared" problem [2]. This mandated the use of an innovative interest-managed communication's architecture [3].

A scalable simulation code capable of 1M entities, known as the Joint Experimentation on Scalable Parallel Processors (JESPP) project [4], grew out of an earlier project named SF Express. [5] The JFCOM experimenters had been constrained in a number of dimensions, *e.g.* numbers, sophistication, realism. Early experiments showed that the new code could scale beyond the 1,000,000 entities [6], but required more computing, *e.g.* a GPU-enhanced cluster.

III. GPGPU ACCELERATION FOR LARGE-SCALE SIMULATIONS

A. Approach

The objective of the effort was to provide stable, distributed and scalable compute resources to JFCOM. Acceleration targets include: line-of-sight calculations, physics-based phenomenology, CFD plume dispersion, data analysis, etc. The GPU has long been a very attractive candidate as an accelerator for computational hurdles, but previous generations of accelerators, *e.g.* Floating Point Systems [7], were for the small market of science and engineering, as opposed to current GPUs that are mass-marketed for gaming.

B. Observations

Even with incomplete utilization of the GPUs, the goal of enabling larger global scale experimentation was exceeded when more than 10M entities were sustained on appropriate terrain with valid phenomenology. (Figure 1)

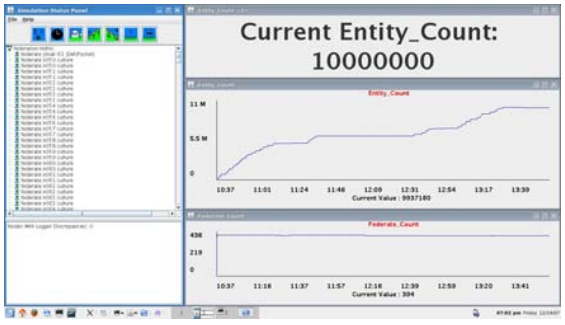

Figure 1. Screen Capture of Ten Million Entity Run

Numerous efforts have been made to increase the entity counts, *e.g.* SF Express [8]. These included the use of the Scalable Parallel Processors (SPP) or Linux clusters, including GPU acceleration [9]. JFCOM teams have made great strides in improving entity behavior models [10 & 11] by adding more realistic entity behaviors. GPUs can be employed to address these issues. The authors found that

1550-6525/10 $26.00 © 2010 IEEE
DOI 10.1109/DS-RT.2010.29

CUDA was easily implemented by journeyman programmers and that, while there were exceptions, the general speedup expected was in the 2X to 3X range for J9 codes

In the quest to advance the broader use of GPUs [12], the new Compute Unified Device Architecture (CUDA) programming language has made GPUs more accessible to programmers [13]. Some potential areas of improvements to the JSAF simulation were identified, e.g. the use of GPU for the route-planning. The successful use of FPGAs as accelerators has been reported [14] and they are installed on compute nodes of some Linux clusters. The raw integer power and reconfigurability of an FPGA is key to cryptography and to fast folding algorithms [15] Clusters could be GPU-enhanced for linear algebra [16] and FFT operations [17].

IV. Transcontinental Systems for Simulation

With the geographic distribution of the computers and the human-in-the-loop participants, as shown in the map below (Figure 2), the authors had to reduce long-haul communications as much as possible:

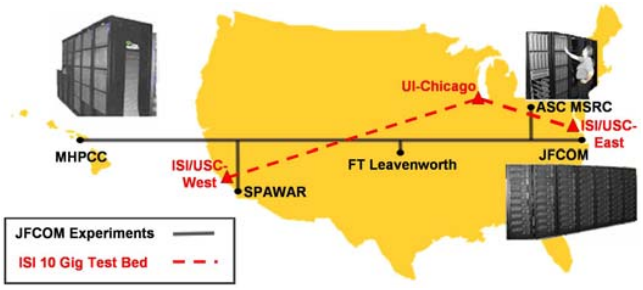

Figure 2. JFCOM Experimentation System and ISI 10 Gig/Sec. Test Bed

A. System Description

This section reports the results of bandwidth tests of interest managed message exchange among processors on three clusters. As the intent was to not interrupt ongoing activities at JFCOM, a separate, but comparable, Wide Area Network (WAN) was established. Previous work [18] had indicated the utility of interest managed communications on cluster meshes, high-bandwidth Local Area Networks (LANs) and lower bandwidth WANs. The current work was specifically investigating high-bandwidth (10 GigaBit per second) WANs with transcontinental distributions.

Interest-limited message exchange was done using ISI's MeshRouter formalism [3]. The main conclusions of the benchmarking studies are 1) Throughput on a single link (client to router, router to router, etc.) is limited to about 320 Mbits/second, reflecting the limitations of the RTI-s [10] communications primitives used in this study. 2) By using multiple routing connections among the participating sites, aggregate bandwidths of 4.8 Gbits/sec were achieved.

The total bandwidth for the aggregate tests represents almost 50% of the nominal WAN bandwidth for the networks used in the tests. While good, it is slightly smaller than rates achieved using simple, net-work performance tests (*i.e.,* "iperf").

B. Interest Managed Routers

The bandwidth experiments were done using the standard ISI MeshRouter formalism for interest-managed communications. A schematic of the MeshRouter is shown in Figure 3.Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".

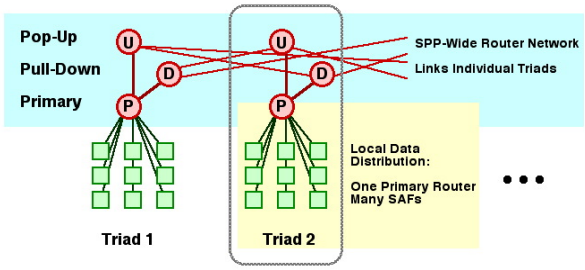

Figure 3. Schematic: MeshRouter Topology

The overall communications scheme consists of collections of processors (labeled "SAFs" in Figure 3) each communicating with a specified "Primary" router (P). Interest-limited message exchange among the various basic "Triads" is done using a network of additional "Pop-Up" and "Pull-Down" routers. The three routers on a triad are instanced as separate objects within a single MeshRouter process. Some details of message management along the various links of are needed in order to assess the results of the bench-mark study. As described in Barrett, *et al.* [3]the MeshRouter software is object-oriented (C++) with daughter classes used to implement key, application specific details, including interest state enumeration, basic message interpretation (i.e., "headers"), and "bits on a wire" communications primitives.

The results reported here use an HLA RTI-s (High Level Architecture, Run Time Infrastructure) implementation for both interest enumeration and the lowest-level communications primitives ("dataflow nodes"). While this has enormous advantages, it does have incompletely understood overheads. Standard RTI-s dataflow implementations exist for both TCP and UDP communications. The results presented here use the TCP implementation.

C. Preliminary Results

The first configurations explored involved a publisher and router at one site (ISI-East) and a subscriber and router at a second site (UIC). This basic configuration was used to explore dependences on various parameters of the basic MeshRouter setup of Figures 3, *e.g.*: packet size within the standard RTI-s data flows and individual message sizes.

TABLE I. Maximum Rates versus Size Parameters

Packet Size(bytes)	10 Kbyte Messages	100 Kbyte Messages	400 Kbyte Messages
8192	-	40 Mbyte/sec	-
65536	35 MByte/sec	40 MByte/sec	30 MByte/sec

The first column in Table 1 specifies a buffer size within the RTI-s software. Throughput did not have strong dependence on this parameter. Attempts with a larger packet size (262144 bytes) resulted in soft-ware failures within the RTI-s libraries on the ISI-E router. The dependence on individual message size reflects known behavior within the full RTI-s package. Smaller messages mean latency on start-up has higher overhead. Very large message sizes incur an overhead from fragmenting and reassembling.

The "optimal" maximum inter-site rate for a single communications link was 320 Mbits/second/link, which was found to be remarkably consistent for all benchmarking tests. Somewhat better rates were ob-served for simple tests will all processes/processors on a single cluster. The intra-processor rates are higher than the inter-processor rates by factors of 2 to 4, the "three-hop" rates (tests 1,2) are about 2/3 of that for the "two hop" tests, and performance is comparable for the inter- and intra-processor configurations.

D. *Wide Area Networks*

The results from the previous sections suggest two general observations: Point to point rates for RTI-s communications are significantly smaller than those within a single cluster and aggregate bandwidth can be increased by exploiting multiple communications paths into and out of individual router processes. These observations suggest that tests across the WAN should involve a rather rich, multiple-router mesh configuration.

A number of variants of the basic configuration were explored, such as the number of distinct interest states (*i.e*., number of processors associated with a single router processes) the number of replicas of the basic "Router plus Associated Pub/Sub" nodes ate each site. Typical performance numbers for a test with eight participating nodes are summarized in Table 2.

TABLE II. PERFORMANCE MEASURES FOR A TYPICAL WAN TEST.

Message Length	**Client BW (bytes/sec)**	**Single MR BW (bytes/sec)**	**Aggregate BW (bites/sec)**
0.4 KByte	3.2 M	16.0 M	1.0 G
0.8 KByte	6.4 M	32.0 M	2.1 G
1.6 KByte	12.8 M	64.0 M	4.1 G
2 KByte	14.3 M	71.5 M	4.6 G
100 KByte	0.8 M	4.0 M	0.3 G

V. DATA MANAGEMENT FOR DISTRIBUTED SIMULATION

These simulations generate terabytes of data that must be effectively managed to be useful to the analyst. The High Level Architecture Object Model Template (HLA OMT) supports simulation interoperability by providing a Federation Object Model (FOM) to formally describe the information interchange (objects, object attributes, interactions, and interaction parameters) within a federation among the federates. Information used by a single federate is defined by the Simulation Object Model (SOM). Often the individual SOMs are mutually incompatible, so standing up a federation typically requires a tedious process modifying the simulation federates to conform to the purposed FOM. Often these measures are invariant with respect to the underlying federation object model.

This section presents a two-layered framework that supports the agile adaptation of analysis tools to specific federations. The top semantic layer provides a modeling framework to capture concepts that analysts tend to use. The concepts include measurements and dimensions, such as object classification, time, and geographic containment. The lower syntactic layer describes how to map the particular federation object models to more abstract semantic concepts. In addition, we show how this approach supports reuse by taking advantage of the hierarchical nature of the object models.

A. *Data Management Organization*

The type of Measure Of Effectiveness questions of interest to analysts are typically not directly captured by simulation loggers. In general analysts are interested in how well higher level mission tasks and objects are satisfied. A MOE is a question or measure, designed to show how well particular tasks are satisfied with respect to a system [19]. MOEs may include percentage of red forces killed/damaged, percentage of blue forces kill/damaged, time take to cross terrain, percentage of forces detected within sensor foot-print, percent of forces detected total, and percentage of detection by sensor type by terrain type by time of day.

B. *Analyst Data Model*

The Sensor/Target Scoreboard provides a visual way of quickly comparing the relative effectiveness of individual sensor platforms and sensor modes against different types of targets [20], [21]. Sensor/Target Scoreboard is a specific instance of the more general multidimensional analysis [22]. In a 2005 I/ITSEC paper the data management and analysis tool was described with the Scalable Data Grid [23].

C. *Distributed Data Management and Results*

The sensors should be able to detect enemy forces and simulating such urban environments requires tremendous amount of distributed computer resources [4]. To work in distributed environments an addi-tional layer is needed to define on top to aggregate multidimensional cubes distributed across different machines. The left-hand side of Figure 4 depicts a single three-dimensional sensor-target-detection status score-cube. It represents only a partial, incomplete view. To generate a complete view, cubes from other simulation federates have to be aggregated. The right-hand side of Figure 4 depicts a tree summing to-gether all the distributed cubes. Again, the associative and commutative

properties of the aggregation operator are used, while the raw data is not sent.

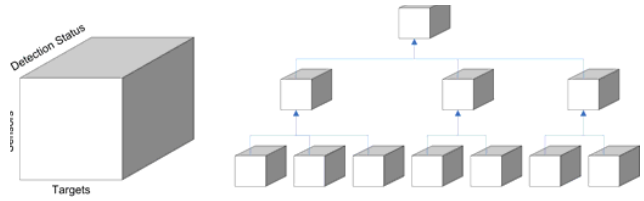

Figure 4. Distributed Data Analysis

VI. Overall Analysis and Conclusions

Simulations continue to demand more and more of technology to deliver faster, more detailed, more sophisticated and more exploitable products. This paper set out three emerging technologies that will likely become mainstays of the simulation community's tool box in the next decade. The work shows new abilities to bring new compute power to bear in order to generate the simulation, to use that power to better analyze the data, to more effectively move the data around the country and more efficiently store the data in such a way as to make it more accessible and more useful.

Acknowledgment

Thanks are due to the excellent staffs at JFCOM, ASC-MSRC and MHPCC. Some of this material is based on research sponsored by the Air Force Research Laboratory under agreement number FA8750-05-2-0204. Other work is based on research sponsored by the U.S. Joint Forces Command via a contract with the Lockheed Martin Corporation and SimIS, Inc. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily rep-resenting the official policies or endorsements, either expressed or implied, of these organizations.

References

[1] Ceranowicz, A. & M. Torpey. 2005. Adapting to Urban Warfare, Journal of Defense Modeling and Simulation, 2:1, January 2005, San Diego, Ca

[2] Brunett, S., & T.D. Gottschalk. 1998. A Large-scale Meta-computing Framework for the ModSAF Real-time Simulation, Parallel Computing:, V24:1873-1900, Amsterdam

[3] Barrett, B. & T.D. Gottschalk. 2004. Advanced Message Routing for Scalable Distributed Simula-tions, in the Proceedings of the 2004 Interservice/Industry Training, Simulation and Education Con-ference, Orlando, Florida.

[4] Lucas, R., & Davis, D., 2003. Joint Experimentation on Scalable Parallel Processors, in the Proceed-ings of the 2003 Interservice/Industry Training, Simulation and Education Conference, Orlando, Flor-ida.

[5] Messina, P. C., S. D. Brunett,., M. Davis, and T. D. Gottschalk. 1997. Distributed Interactive Simula-tion for Synthetic Forces, In Mapping and Scheduling Systems, International Parallel Processing Symposium, Geneva

[6] Wagenbreth, G., & . Davis, D.M., 2005, Enabling 1,000,000-Entity Simulations on Linux Clusters in the Proceedings of 2005 Winter Simulation Conference, Orlando, FL.

[7] Charlesworth, A., & J. Gustafson, J. 1986. Introducing Replicated VLSI to Supercomputing: the FPS-164/MAX Scientific Computer, in IEEE Computer, 19:3, pp 10-23, March 1986

[8] Brunett, S., Messina, P.C., Gottschalk, T.D., Davis, D.M., & Kesselman, C., 1998, Implementing Dis-tributed Synthetic Forces Simulations in Metacomputing Environments, The Seventh Heterogeneous Computing Workshop, Orlando, FL.

[9] Wagenbreth, G., Lucas, R.F. & . Davis, D.M., 2007, A GPU-Enhanced Cluster for Accelerated FMS, in the Proceedings of the HPCMP Users Group Conference, Pittsburgh, Pennsylvania

[10] Ceranowicz, A., M. Torpey, B. Helfinstine, J. Evans, & J. Hines. 2002. Reflections on Building the Joint Experimental Federation, in the Proceedings of the 2002 Interservice/Industry Training, Simula-tion and Education Conference, Orlando, Florida.

[11] Ceranowicz, A., M. Torpey, B. Helfinstine, J. Evans, & J. Hines. 2006. Reflections on Building the Joint Experimental Federation, in the Proceedings of the 2006 Interservice/Industry Training, Simula-tion and Education Conference, Orlando, Florida.

[12] Lastra, A., M. Lin, and D. Minocha, 2004. ACM Workshop on General Purpose Computations on Graphics Processors.

[13] Buck, I., 2007. GPU Computing: Programming a Massively Parallel Processor, International Sympo-sium on Code Generation and Optimization, San José, California

[14] Linderman R. W., M. H Linderman,., and C-S. Lin. 2005. FPGA Acceleration of Information Man-agement Services, 2005 MAPLD International Conference, Washington, DC

[15] Frigo., J., Palmer, D., Gokhale, M., and M. Popkin-Paine, M. 2003, Gamma-ray pulsar detection us-ing reconfigurable computing hardware, 11th Annual IEEE Symposium on Field-Programmable Cus-tom Computing Machines FCCM.

[16] Fatahalian, K., Sugerman, .J. & Hanrahan, P., 2004. Understanding the efficiency of GPU algorithms for matrix-matrix multiplication, Workshop on Graphics Hardware, Eurographics/SIGGRAPH

[17] Sumanaweera, T. and D. Liu. 2005. Medical Image Reconstruction with the FFT, in GPU Gems 2, M. Pharr, Ed. Boston: Addison-Wesley

[18] Gottschalk, T., P. Amburn, & D. Davis. 2005. Advanced Message Routing for Scalable Distributed Simulations, The Journal of Defense Modeling and Simulation, Vol 2. Issue 1:17-28, San Diego, California

[19] Gertner, A. S. & Webber B. L., A Bias towards Relevance: Recognizing Plans where Goal Minimi-zation Fails. AAAI/IAAI, Vol. 2 1996: 1133-1138

[20] Graebener, R., G. Rafuse, , R. Miller., & L-T. Yao. 2003. The Road to Successful Joint Experimen-tation Starts at the Data Collection Trail in the Proceedings of the 2003 Interservice/Industry Train-ing, Simulation and Education Conference, Orlando, Florida.

[21] Graebener, R., G. Rafuse, , R. Miller., & L-T. Yao. 2004. The Road to Successful Joint Experimenta-tion Starts at the Data Collection Trail—Part II in the Proceedings of the 2003 Interservice/Industry Training, Simulation and Education Conference, Orlando, Florida.

[22] Kimball, R., L. M. Reeves, M. Ross, & W. Thornwaite. 1998. The Data Warehouse Lifecycle Tool-kit. Hoboken, New Jersey: Wiley.

[23] Yao, K.-T., & Wagenbreth, G. 2005. Simulation Data Grid: Joint Experimentation Data Management and Analysis. In the Proceedings of the 2005 Interservice/Industry Training, Simulation and Educa-tion Conference, Orlando, Florida.

[24] Uschold, M. & Grüninger, M., 2004, Ontologies and Semantics for Seamless Connectivity. SIGMOD Record 33(4): 58-64